SS Mini project – Absolute loader (Viva)

★ Loading
→ bring obj file into memory } Absolute loader

★ Relocation
→ modify obj file such that it can be loaded to a memory address different from what's originally specified in obj file.

} Relocating loader aka Relative loader

★ Linking
→ combine multiple obj files & allow references between them } Linker

} Linking loader

★ 1 block = 8 hexadecimal digits = 8 × 4 bits = 32 bits.

★ Each line in output.mem has 4 blocks. (each line indicate contiguous range of blocks in memory)

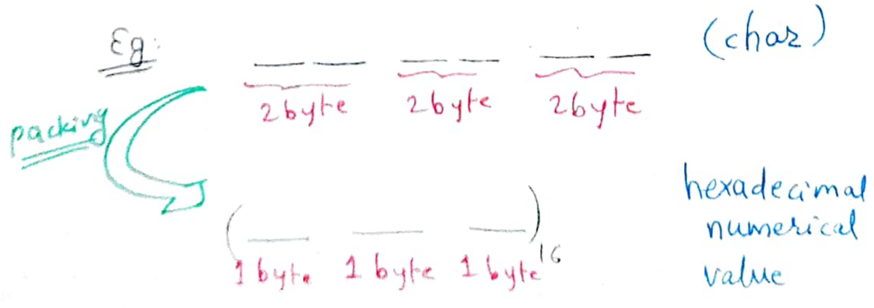
★ 6-digit hexadecimal object code ⇒ occupies 3 bytes.

★ when loader reads object codes, it does NOT read them as hexadecimal numbers, but reads them as characters
↓
1 char = 1 byte.

∴ 6 digits in object code ⇒ 6 bytes for 6 chars ⇒ inefficient

∴ Loader "packs" pairs of chars into their hexadecimal equivalent.

Eg:



Eg: "14" ⟶ $(14)_{16}$
(chars) (2bytes) (hex value) (1 byte) > (0001 0100)

★ End record signifies the memory location from where program execution must start from, & places a pointer at that location. (this location is basically the starting address, 0x0000h, i.e, $(00)_{16}$).

★ Algorithm for absolute loader :

```
begin
    read Header record
    verify program name and length
    read first Text record
    while record type ≠ 'E' do
        begin
            {if object code is in character form, convert into
             internal representation, i.e, hexadecimal.}
            move object code to specified memory location.
            read next object program record.
        end
    jump to address specified in End record.
end.
```

★

$$\begin{tikzcd}
 & \boxed{\text{Loaders}} \arrow[dl] \arrow[d] \arrow[dr] & \\
\text{Absolute loaders} & \text{Relocating loader} \arrow[dl] \arrow[dr] & \text{Linking loader} \\
\text{for SIC} \arrow[d] & & \text{for SICXE} \arrow[d] \\
\text{use Relocating bits} & & \text{use modification records.}
\end{tikzcd}$$

★ BOOTSTRAP LOADER

- It loads the first program to be run by the computer (operating system)
- The bootstrap loader itself is loaded at memory location $(00)_{16}$ (0x0000)
- Bootstrap loader is an absolute loader which loads the OS at $(80)_{16}$.